Cluster

From Eigenvector Documentation Wiki
Cluster

Contents

Purpose

Hierarchical Cluster Analysis with dendrograms.

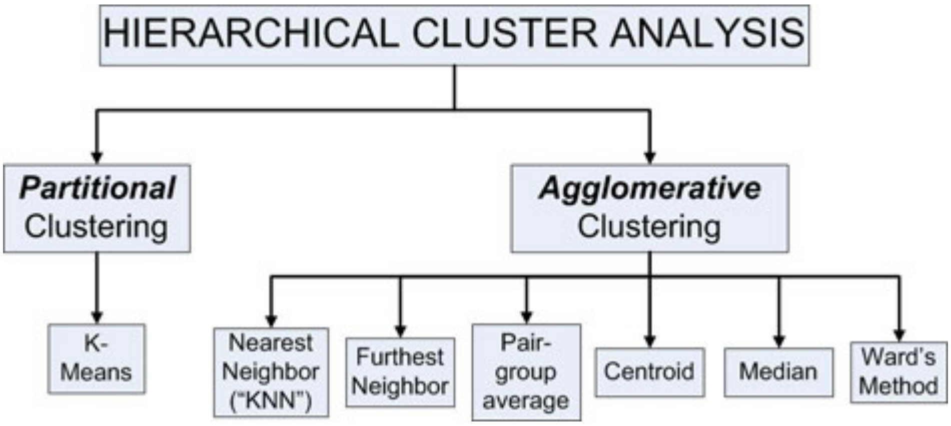
Synopsis

[results,fig,distances] = cluster(data,*labels,options*)
[results,fig,distances] = cluster(data,*options*)
cluster % launches an Analysis window with cluster as the selected method.

Description

cluster performs unsupervised hierarchical cluster analysis (HCA) on the input data, using one of several possible clustering methods.

Cluster Analysis methods can be classified into two main categories: **agglomerative** and **partitional**. Agglomerative methods begin with each object being it's own cluster, and progress by combining (agglomerating) existing clusters into larger ones. Partitional methods start with a single cluster containing all objects, and progress by dividing existing clusters into smaller ones.

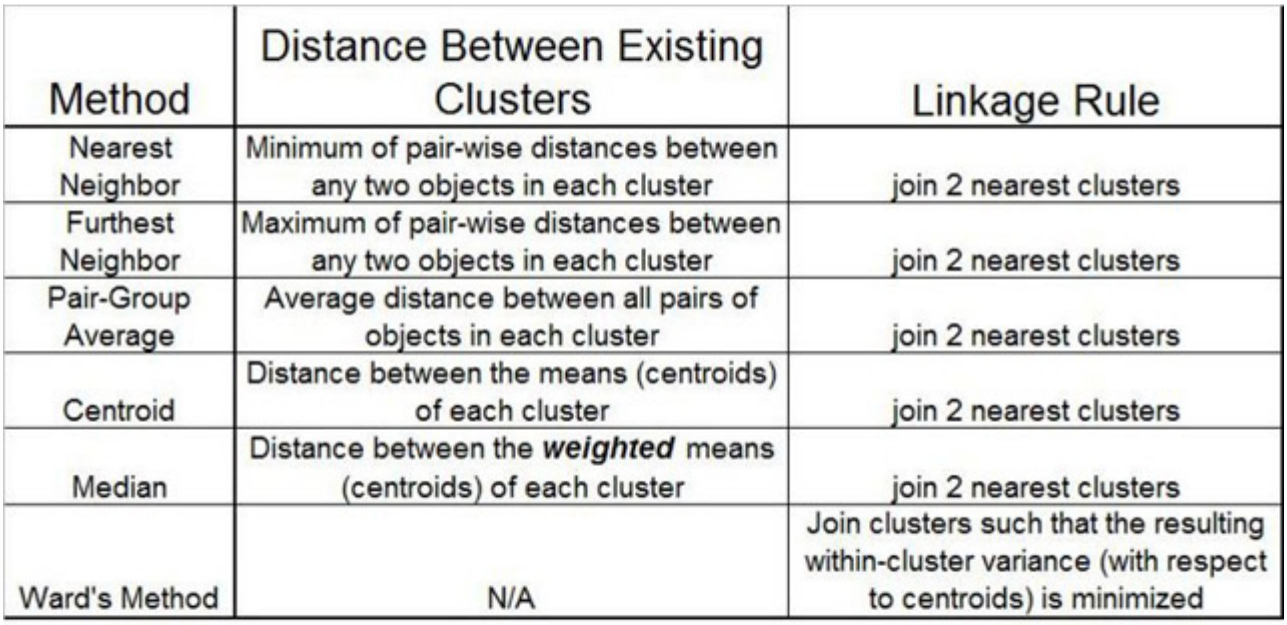


The *cluster* function enables the use of six different agglomerative methods (Nearest Neighbor, Furthest Neighbor, Pair Group Average, Centroid, Median and Ward's Method) and one partitional method (K-means).

All clustering methods require the specification of a **distance measure** to be used to indicate distances between objects, and subsequently between clusters, during method operation. First, one could use either the original X-variables or PCA scores to determine the distance. The usage of PCA scores can provide colinearity and noise-reduction benefits, but requires the specification of the appropriate number of PCs. Additionally, given the input variables (X-variables or PC scores) one can then choose either the Euclidean or the Mahalanobis distance to complete the definition of the distance measure. The use of Mahalanobis distance allows one to account for dominant multivariate directions in the data when performing cluster analysis.

The **K-means** partitional clustering method starts with a random selection of K objects that are to be used as cluster targets, where K is determined *a priori*. During each cycle of this clustering method, the remaining objects are assigned to one of these clusters, based on distance from each of the K targets. New cluster targets are then calculated as the means of the objects in each cluster, and the procedure is repeated until no objects are re-assigned after the updated mean calculations.

All six of the agglomerative methods start with each object defining it's own cluster. Then, each cycle of the method involves the calculation of all possible inter-cluster distances based on the method's definition of inter-cluster distance (for 5 of the 6 methods), followed by the joining of two clusters based on the method's linkage rule. The table below summarizes the inter-cluster distance definitions and linkage rules for the six agglomerative methods in PLS_Toolbox.



Method	Distance Between Existing Clusters	Linkage Rule
Nearest Neighbor	Minimum of pair-wise distances between any two objects in each cluster	join 2 nearest clusters
Furthest Neighbor	Maximum of pair-wise distances between any two objects in each cluster	join 2 nearest clusters
Pair-Group Average	Average distance between all pairs of objects in each cluster	join 2 nearest clusters
Centroid	Distance between the means (centroids) of each cluster	join 2 nearest clusters
Median	Distance between the ***weighted*** means (centroids) of each cluster	join 2 nearest clusters
Ward's Method	N/A	Join clusters such that the resulting within-cluster variance (with respect to centroids) is minimized

Short descriptions of each of these methods are given below:

- **Nearest Neighbor** : The distance between any two clusters is defined as the **minimum** of all possible pair-wise distances of objects between the two clusters; the two clusters with the minimum distance are joined together. This method tends to perform well with data that form elongated "chain-type" clusters.
- **Furthest Neighbor** : The distance between any two clusters is defined as the **maximum** of all possible pair-wise distances of objects between the two clusters; the two clusters with the minimum distance are joined together. This method tends to perform well with data that form "round", distinct clusters.
- **Pair-Group Average** : The distance between any two clusters is defined as the average distance of all possible pair-wise distances between objects in the two clusters; the two clusters with the minimum distance are joined together. This method tends to perform equally well with both "chain-type" and "round" clusters.
- **Centroid** : The distance between any two clusters is defined as the difference in the multivariate means (centroids) of each cluster; the two clusters with the minimum distance are joined together.
- **Median** : The distance between any two clusters is defined as the difference in the **weighted** multivariate means (centroids) of each cluster, where the means are weighted by the number of objects in each cluster; the two clusters with the minimum distance are joined together. This method might perform better than the Centroid method if the number of objects is expected to vary greatly between clusters.
- **Ward's Method** : This method does not require calculation of the cluster centers; it joins the two existing clusters such that the resulting pooled within-cluster variance (with respect to each cluster's centroid) is minimized.

Inputs

- *data*: input data, class double or dataset
- *labels* (optional input): used to put labels on the dendrogram plots. For data *M* by *N* then *labels* must be a character array with *M* rows. When *labels* is not specified and data is

class "double", the dendrogram is plotted using sample numbers. When *labels* is not specified and data is a DataSet object, the dendrogram is plotted using sample labels included in the DataSet object. If the DataSet labels field is empty it will use sample numbers.

Note: Calling cluster with no inputs starts the graphical user interface (GUI) for this analysis method.

Outputs

- **results** = a structure containing results of the clustering (defined below)
- **fig** = the handle to any plot created.
- **distances** = the matrix of sample-to-sample distances

The **results** output contains the following fields:

- **dist** : the distance threshold at which each cluster forms.
- **class** : the classes of each sample (columns of class) for each distance (rows of class).
- **order** : the order of the samples which locates similar samples nearest to each other (this is the order used for the plots).
- **linkage** : a table of linkages where each row indicates a linkage of one group to another. Each row in the matrix represents one group. The first two columns indicate the sample or group numbers which were linked to form the group. The final column indicates the distance between linked items. Group numbers start at m+1 (where m is the number of samples in the input data matrix) thus, row j of this matrix is group number m+j. This matrix can be used with the statistics toolbox dendrogram function.

The (results.class) matrix can be used with the (results.dist) matrix to determine clusters of samples for any distance using:

```
results   = cluster(data);                      %do cluster
ind       = max(find(results.dist<threshold));  %user-desired threshold
thisclass = results.class(ind,:);               %grab arbitrary classes
```

cluster also outputs a dendrogram showing the sample distances. This dendrogram is interactive, in that one can place a vertical cursor at a specified distance in order to view the corresponding cluster groupings.

The **distances** matrix is a square matrix (size samples x samples) containing all the inter-sample distances on which the clustering was based (not available with algorithm = 'kmeans'.)

Options

Optional input (*options*) is a structure array with the following fields:

- **plots**: ['none' | {'final'}] governs level of plotting
- **algorithm**: ['knn' | 'fn' | 'med' | 'avgpair' | 'cnt' | 'kmeans' | {'ward'}] specifies the cluster method
    - knn : K-Nearest Neighbor

    - fn : Furthest Neighbor
    - avgpair : Average Paired Distance
    - med : Median
    - cnt : Centroid
    - ward : Ward's Method {DEFAULT}
    - kmeans : K-means

- **preprocessing**:{[ ]} Preprocessing structure or keyword (see PREPROCESS),
- **pca**:[{'off'} | 'on'] if 'on' then clustering is done on the PCA scores instead of the original X-variables
- **ncomp**:[ ] number of PCA factors to use {default = [ ], and the user is prompted to select the number of factors from the SSQ table}
- **mahalanobis**:[{'off'} | 'on'] if 'on' then a Mahalanobis distance measure is used; if not, then Euclidean distance is used
- **slack**:[0] integer number indicating how many samples can be "overridden" when two class branches merge. If the smaller of the two classes has no more than this number of samples, the branch will be absorbed into the larger class. This feature is only valid when classes are supplied in the input data. A value of 0 (zero) disables this feature.
- **distance**: [{'euclidean'} | manhattan] governs if Euclidean distance or Manhattan distance will be used in the cluster method. Note: When using PCA scores (i.e. options.pca is set to 'on'), only Euclidean distance can be used.

See Also

analysis, corrmap, dbscan, dendrogram, gcluster, knn, knnscoredistance, simca

Retrieved from "http://wiki.eigenvector.com/index.php?title=Cluster&oldid=9433"

- This page was last modified on 27 January 2017, at 13:10.
- See This Page Online